2

Teknik Audit Berbantuan Komputer

TABK (Teknik Audit Berbantuan Komputer) secara sederhana dapat didefinisikan sebagai penggunaan komputer untuk melaksanakan prosedur-prosedur audit dalam rangka mencapai tujuan audit tertentu secara efisien dan efektif. Dari pengertian ini, yang perlu ditekankan adalah TABK bukanlah satu-satunya prosedur audit. TABK bukan dimaksudkan untuk menggantikan prosedur audit lain. TABK hanya digunakan manakala bukti audit tidak bisa diperoleh dan dievaluasi secara manual. Dalam praktiknya, TABK akan digunakan bersama-sama dengan prosedur audit konvensional seperti wawancara dan observasi fisik.

2.1 Jenis TABK

Jenis TABK dibedakan menurut objek yang akan menjadi titik berat pengujiannya, yaitu:

- TABK untuk menguji **data**
- TABK untuk menguji **sistem**

Adapun ragam TABK untuk kedua kelompok tersebut dapat dilihat dalam tabel berikut.

Tabel 2.1 Jenis TABK

TABK untuk menguji data	TABK untuk menguji sistem
Data file interrogation (DFI)	Test data
Embedded audit module (EAM)	Parallel simulation
	Integrated test facilities
	Program code comparison
	Program code review

2.1.1 TABK untuk Menguji Data

Dua teknik utama yang sering digunakan untuk menguji data, yaitu interogasi data *(data file interrogation)* dan penanaman modul audit *(embedded audit modules)*.

Data File Interrogation (DFI)

DFI merupakan teknik untuk me-review data dan informasi dalam file komputer menggunakan software audit. DFI mengandalkan kecepatan dan keandalan komputer agar auditor dapat menangani data yang volumenya sangat banyak. Operasi yang sering dilakukan auditor terkait data, meliputi:

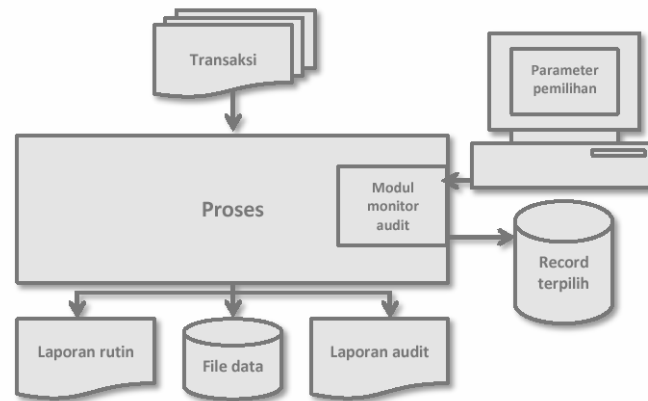
- memilih *record* yang sesuai dengan kriteria yang ditetapkan;
- mencetak *record* terpilih untuk pengujian lebih lanjut;
- mencetak *totals* dan *subtotals* dari sebuah file akuntansi;
- menyajikan data berdasarkan nilai tertentu (stratifikasi data);
- mencari transaksi-transaksi ganda (duplikasi);
- mencari data yang hilang dari urutan tertentu (gap);
- membandingkan isi dari dua atau lebih file dan mencetak *record* yang sesuai *(matches)* atau yang tidak sesuai *(exception);*
- mengurutkan dan menggabungkan file untuk pengujian lanjutan, seperti pembandingan file dan analisis gap.

DFI juga digunakan dalam penerapan berbagai jenis *sampling,* yaitu:

- random sampling;
- interval sampling;
- monetary unit sampling;
- cell monetary unit sampling.

DFI biasanya dilakukan menggunakan software yang secara khusus dibuat untuk mengakses dan mengevaluasi data, seperti *Audit Command Language (ACL)* dan *Interactive Data Evaluation and Analysis (IDEA)*. Microsoft Excel juga bisa digunakan untuk keperluan ini, meskipun dengan segala keterbatasan yang dimiliki-kinya.

Embedded Audit Modules (EAM)

EAM pada dasarnya merupakan sebuah teknik yang dilakukan dengan cara "menanamkan" suatu kode program tertentu di dalam program utama yang digunakan oleh klien. Kode program ini dibuat sedemikian rupa agar auditor dapat memperoleh data yang sesuai dengan parameter atau kriteria yang telah ditentukan. Cara ini dilakukan apabila auditor menghadapi data yang sangat banyak dan auditor hanya menginginkan data tertentu saja. Modul yang ditanamkan akan menguji setiap transaksi yang masuk ke dalam sistem. Jika transaksi tersebut memenuhi kriteria yang telah ditetapkan, maka transaksi tersebut akan "dicatat" oleh modul tadi sebelum diteruskan untuk diproses lebih lanjut dalam sistem tersebut. Catatan yang berhasil dihimpun oleh modul tersebut secara periodik akan dipindai, dianalisis, dan dilaporkan untuk memperoleh tindak lanjut. Gambaran EAM kurang lebih seperti berikut ini.

Gambar 2.1 Embedded Audit Modules

Meskipun ide dasarnya sederhana, namun dalam kenyataannya penggunaan EAM bisa dibilang rumit. Auditor tidak bisa sembarangan menggunakan teknik ini karena modul yang ditanam bisa mengganggu sistem utama. Untuk itu, penggunaan EAM harus benar-benar dikonsultasikan dengan klien untuk memperoleh persetujuan karena terkait alasan integritas data dan sistem yang mereka miliki. Selanjutnya, dalam praktiknya, auditor harus melibatkan ahli yang mengerti betul mengenai bahasa pemrograman komputer, khususnya yang terkait dengan program yang digunakan klien. Apabila EAM digunakan tanpa perencanaan dan otorisasi, bisa-bisa modul yang ditanam auditor malah merusak sistem klien.

2.1.2 TABK untuk Menguji Pengendalian Sistem

Di samping untuk menguji data, TABK juga dimanfaatkan untuk menguji pengendalian sistem. Pengujian atas pengendalian sistem perlu dilakukan dengan teknik khusus yang melibatkan penggunaan komputer karena auditor tidak akan melakukannya secara manual. Teknik-teknik yang biasa digunakan untuk menguji pengendalian sistem, meliputi *test data, integrated test facility, parallel simmulation, program code comparison,* dan *program code review.*

Test Data

Dalam pengujian pengendalian sistem, adakalanya auditor ingin mengetahui apakah sebuah program aplikasi telah memproses data dengan benar, mengeluarkan *output* yang benar sesuai yang direncanakan. Auditor mungkin melakukannya dengan membandingkan antara input dan output, dan dari sana ia akan menilai apakah output tersebut memang benar, dan hal ini akan menegaskan bahwa program yang diujinya memang bekerja dengan baik.

Alternatif lain, auditor juga dapat menggunakan seperangkat data yang disiapkan untuk menguji logika program, seperti yang dilakukan pemrogram dalam menguji aplikasi yang dibuatnya pada tahap pengembangan. Dalam hal ini, yang dilakukan auditor adalah memasukkan data yang telah disiapkan tersebut, biasanya disebut *dummy data,* ke dalam program dan membandingkannya dengan output yang dihasilkan program tersebut. Cara ini disebut dengan *test data.*

Test data biasanya digunakan untuk menguji program tambahan yang baru atau program yang akan menghasilkan output yang tidak dapat dengan mudah diprediksi atau direkonsiliasi dengan inputnya. Teknik ini dapat digunakan untuk menguji dan memverifikasi:

- proses validasi input;
- kemampuan mendeteksi kesalahan;
- logika pemrosesan dan perhitungan;
- keakuratan laporan;
- prosedur manual terkait program.

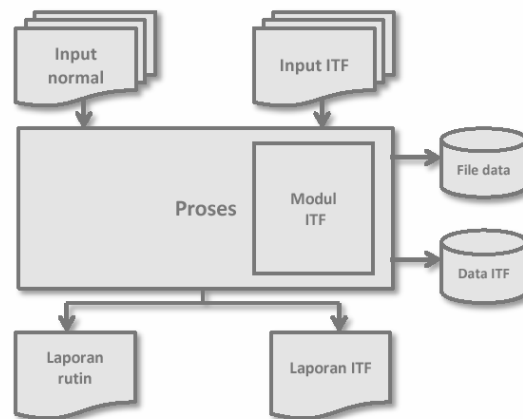
Sepintas, teknik ini mudah dilaksanakan dan bisa menjadi solusi ideal untuk mengecek operasi sebuah program. Kelebihan yang diperoleh dengan teknik ini, yaitu:

- membutuhkan pengetahuan teknis terbatas;
- relatif mudah dilaksanakan;
- membantu auditor memahami bagaimana sistem bekerja.

Namun demikian, di balik kemudahan tersebut, terdapat beberapa hal yang perlu diperhatikan ketika menggunakan teknik *test data*. Auditor harus menyadari hal-hal berikut.

- *Test data* hanya berlaku untuk program yang saat itu sedang diuji;
- Apabila program yang diuji masih dalam tahap pengembangan, auditor tidak akan memperoleh jaminan bahwa program yang akhirnya digunakan secara *live* adalah program yang diuji, dan bahwa antarmuka *(interfaces)* pada program yang diuji akan sama dengan program yang digunakan secara *live*. Untuk itu, auditor harus memastikan bahwa klien memiliki struktur manajemen dan pengendalian untuk setiap perubahan *(change control)* program. Hal ini dimaksudkan agar software yang telah lolos pengujian kualitas dalam tahap pengembangan dilindungi dari kemungkinan perubahan yang tidak sah. Dengan demikian, ada keyakinan bahwa program yang dipakai adalah program asli *(true copy)* dan tidak akan diubah setelah tahap pengujian selesai dilakukan.
- Auditor tidak diperbolehkan menguji program dengan data *live,* yang mengandung berbagai kemungkinan situasi yang mungkin tidak dapat diantisipasi pada tahap pengembangan. Oleh karena itu, auditor harus merancang *test data* untuk menguji berbagai macam fungsi sehingga ia akan dapat menentukan apa saja yang seharusnya dikerjakan dan apa saja yang tidak boleh dikerjakan oleh program yang sedang diujinya.

Integrated Test Facility (ITF)

ITF adalah teknik audit yang digunakan untuk menguji sebuah sistem yang kompleks. Teknik ini dilakukan dengan membuat sebuah fasilitas "tandingan" yang lengkap dalam sebuah departemen atau cabang perusahaan rekaan *(dummy)* yang juga menjalankan fungsi akuntansi normal seperti departemen aslinya. Perusahaan perbankan biasanya menggunakan teknik ini untuk keperluan audit maupun mendidik calon-calon *teller*.

Gambar 2.2 Integrated Test Facility

Sekali fasilitas tersebut dibuat, metode audit yang dilakukan hampir sama dengan *test data*. Auditor bank tersebut akan diberi satu terminal tersendiri, dan dengan fasilitas ini ia dapat menciptakan pelanggan baru pada *master file* serta dapat mencatat penyetoran, menarik dana, dan berbagai tranksaksi lainnya, tentunya dengan data *dummy* yang diciptakannya. Transaksi tersebut akan diproses dan dilaporkan secara normal dengan program atau sistem yang tengah digunakan bank. Perbedaan utamanya adalah bahwa pengecekan secara signifikan harus dilakukan untuk mencegah kerusakan sistem karena penggunaan *test data*. Untuk itu, transaksi yang dilakukan melalui ITF tidak boleh:

- mengganggu data akuntansi manajemen bank;
- merusak statistik perbankan pemerintah;
- melakukan transfer dana ke atau dari rekening *live*.

Adapun kelebihan penggunaan ITF adalah:

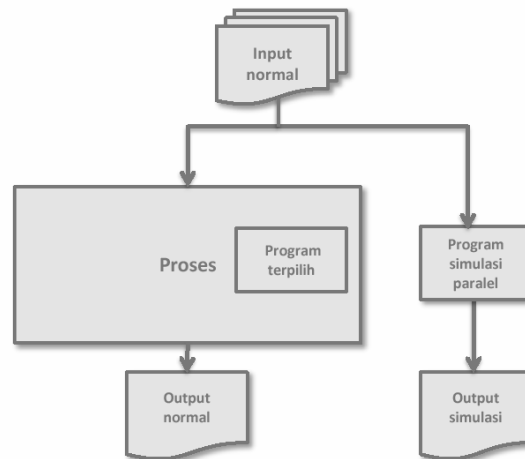
- memudahkan pengujian komprehensif atas sistem secara *live;*
- pengujian dapat dilakukan kapan saja, tanpa diketahui staf lainnya;

- biaya yang diperlukan kecil untuk sekali fasilitas tersebut dibuat;
- memberikan bukti primer dari sistem yang bekerja secara benar;
- dapat digunakan untuk menguji sistem dan melatih karyawan baru.

Terlepas dari berbagai kemudahan dan kelebihan tersebut, penggunaan ITF harus ditentukan pada saat pengembangan sistem sehingga dampak terhadap sistem secara menyeluruh bisa dikalkulasi dari awal dan pengembangannya dapat diintegrasikan sekaligus dengan sistem utama.

Parallel Simmulation

Tujuan simulasi paralel adalah untuk menghasilkan sebuah program yang secara independen akan membuat simulasi sebagian aplikasi dalam program yang diuji. Sebagai contoh, anggaplah Anda ingin membuktikan bahwa penghitungan bunga telah berjalan dengan benar, namun karena berhadapan dengan data yang begitu banyak, Anda tidak dapat melakukannya dengan mudah. Untuk alasan ini, Anda kemudian membuat sebuah aplikasi sederhana untuk menghitung bunga berdasarkan transaksi riil. Setelah memasukkan semua data transaksi riil, Anda kemudian membandingkan hasil perhitungan bunga dengan aplikasi yang Anda buat dengan hasil perhitungan program yang Anda uji. Dengan pembandingan ini, Anda akan mengetahui apakah penghitungan bunga telah dilakukan dengan benar.

Pengujian dengan teknik ini tidak serumit yang dibayangkan karena simulasi yang dibuat biasanya hanya menyangkut satu atau dua aspek dari program yang diuji, dan biasanya lebih kecil dan tidak kompleks. Meskipun demikian, seperti *test data,* Anda harus ingat bahwa hasil yang diperoleh hanya terkait dengan pengujian yang Anda lakukan. Suatu saat, program yang saat ini Anda uji barangkali akan diubah, dan untuk alasan ini, Anda harus melakukan kembali pengujian serupa.

Gambar 2.3 Parallel Simulation

Program Code Comparison

Teknik ini melibatkan pembandingan dua versi sebuah program untuk mengetahui perbedaan di antara keduanya. Proses pembandingan kode program ini bisa dilakukan dengan program-program bantu *(utilities)* tertentu. Dengan menggunakan *utilities* ini, semua perbedaan kode antara dua program dapat diidentifikasi kemudian dicek untuk memastikan bahwa perubahan-perubahan yang dilakukan telah diotorisasi.

Program Code Review

Teknik ini melibatkan pengujian secara rinci atas kode program. Oleh karena itu, teknik ini hanya bisa dilakukan oleh orang-orang yang memiliki keterampilan pemrograman aplikasi dan pengetahuan yang mendalam mengenai spesifikasi aplikasi tersebut. Seperti teknik lainnya, review kode program juga tidak memberikan jaminan bahwa kode yang sedang di-review merupakan kode untuk program yang sedang dijalankan secara *live,* kecuali klien memiliki prosedur atau pengendalian yang baik untuk setiap perubahan program *(change controls)*.

2.2 Software untuk TABK

Program-program yang digunakan oleh auditor untuk menginterogasi file, yang biasa disebut sebagai *software audit,* sangat beragam. Namun, bila dibuat klasifikasi, ragam audit software tersebut akan dapat dikelompokkan menjadi *data retrieval packages, manufacturer's utility software, enquiry programme,* dan *programming language.*

2.2.1 Data Retrieval Packages

Software audit kelompok ini dibuat secara khusus untuk menangani pengolahan data yang biasa dilakukan dalam audit. Software yang termasuk pada kategori ini, antara lain ACL *(audit command language)* dan IDEA *(interactive data extraction and analysis).* Fasilitas yang diberikan oleh software ini, meliputi:

- Menyortir dan mengindeks files;
- Mengekstraksi *record* tertentu dengan kriteria yang ditetapkan;
- Analisis statistik;
- Deteksi duplikasi dan gap;
- Pembandingan file;
- Stratified sampling.

Dengan menggunakan software seperti ACL dan IDEA, semua aktivitas tersebut dapat dilakukan dengan mudah karena sudah menjadi menu yang sudah terpasang *(built-in)* dalam software tersebut. Oleh karena itu, untuk menggunakannya, auditor tidak perlu memiliki pengetahuan mengenai pemrograman khusus.

2.2.2 Manufacturer's Utility Software

Hampir semua perusahaan pembuat komputer selalu menyediakan program *utility,* sebagai tambahan dari sistem operasi. Meskipun tidak dirancang khusus untuk auditor, namun program-program ini sangat bermanfaat untuk memanipulasi data agar sesuai dengan kebutuhan auditor. Operasi yang biasa dilakukan

dengan program *utility* ini mencakup mengekstraksi data spesifik dari database, mengurutkan *(sorting),* menggabung file *(merging and joining),* dan mengambil *record* file hexadecimal untuk menguji isi file lebih dekat.

2.2.3 Enquiry Programme

Di pasaran banyak beredar program akuntansi keuangan yang dibangun oleh perusahaan pengembang software, seperti ORACLE, Sun Systems, dan SAP. Program-program tersebut biasanya mengikutsertakan sebuah paket program *enquiry* yang dapat digunakan untuk memilih item-item tertentu guna diunduh ke sebuah *personal computer* (PC).

2.2.4 Programming Language

Utility programs, enquiry programs, dan *audit packages* mengotomatisasi banyak fungsi pemrograman, seperti *opening, closing, sorting, comparing files, automatic totalling,* dan *stratification.* Namun demikian, kadang kala auditor ingin membuat software audit sendiri menggunakan sebuah bahasa pemrograman, terutama ketika kompleksitas pemrosesan data melebihi kemampuan software audit, seperti ACL dan IDEA. Hanya saja, penggunaan bahasa pemrograman untuk membuat aplikasi untuk keperluan audit secara spesifik membutuhkan waktu dan kadang kurang fleksibel dalam penggunaannya. Belum lagi adanya risiko kesalahan pemrograman yang dapat berakibat pada tidak berfungsinya software yang dibuat. Untuk itu, pengetesan terhadap aplikasi yang dibuat perlu dilakukan, jika tidak maka akan dapat menghasilkan data yang tidak benar.

Adapun bahasa pemrograman yang biasa digunakan mencakup ‘C’, COBOL, BASIC, dan PL/I. Namun dalam TABK level ini, auditor perlu dilatih untuk menguasai bahasa pemrograman, desain program, dan penggunaan software dalam lingkungan sistem operasi tertentu, seperti Unix. Mengingat tingkat kesulitannya, TABK dengan bahasa pemograman hanya bisa dilakukan oleh mereka yang benar-benar ahli.

2.3 Pertimbangan dalam Menerapkan TABK

Seperti dinyatakan dalam paragraf sebelumnya, TABK bukanlah segala-galanya dan bukanlah satu-satunya prosedur yang harus ditempuh ketika Anda berhadapan dengan klien yang memiliki sistem informasi berbasis komputer. Dalam sebuah penugasan audit, dengan berbagai pertimbangan, Anda bisa memutuskan untuk menggunakan TABK atau tidak. Secara ringkas, faktor-faktor yang dapat menjadi pertimbangan penggunaan TABK adalah sebagai berikut.

- Apakah tujuan audit yang akan dicapai?
- Apakah jenis data (bukti audit) yang dibutuhkan untuk mencapai tujuan tersebut?
- Apakah data yang dimiliki oleh auditee?
- Apakah data tersedia dalam format yang sesuai? Jika tidak, apakah data tersebut dapat disajikan dalam format yang sesuai?
- Apakah sumber informasi yang tersedia?
- Apakah TABK yang seharusnya digunakan? Apa keunggulan dan kelemahannya?
- Apakah tingkatan keterampilan yang tersedia dalam tim audit?
- Berapakah biaya yang harus dikeluarkan bila menggunakan TABK?

Pertanyaan pertama untuk memutuskan apakah Anda akan menggunakan TABK atau prosedur yang lain tentu saja menyangkut tujuan audit. Anda tentu ingat bahwa pencapaian tujuan audit dapat ditempuh dengan berbagai prosedur audit, tergantung pada ketersediaan dan jenis bukti yang relevan dengan tujuan itu. Apabila bukti audit relevan yang harus Anda peroleh hanya tersedia dalam wujud elektronis, tentu TABK menjadi pilihan yang harus dipertimbangkan. Namun, bila bukti tersebut secara faktual hanya bisa diperoleh dengan prosedur konvensional, seperti wawancara, dokumen, dan penghitungan fisik, maka Anda tidak perlu melirik TABK sebagai prosedur audit. Apabila hal terakhir ini yang terjadi, segera lupakan pertanyaan berikutnya.

Jika prosedur audit yang harus ditetapkan melibatkan TABK, pertanyaan selanjutnya yang harus dijawab adalah soal keter-

sediaan data pada auditee berikut format datanya. Anda harus memastikan bahwa data yang Anda butuhkan untuk mencapai tujuan audit yang telah ditetapkan bisa dipenuhi oleh auditee. Anda juga perlu mempertimbangkan format data yang dimiliki. Pertimbangan ini diperlukan karena kemampuan Anda mengolah data akan sangat bergantung pada teknologi yang Anda miliki. Jika Anda memiliki berbagai teknologi yang diperlukan untuk berbagai jenis TABK, ragam format data tentu tidak akan menjadi masalah. Namun, jika selama ini hanya Microsoft Excel yang Anda miliki, maka Anda harus mengupayakan agar data yang akan diperoleh nantinya bisa terbaca oleh aplikasi tersebut.

Setelah pertimbangan soal data terjawab, Anda harus memastikan sumber-sumber perolehan data. Anda harus yakin bahwa auditee memiliki sumber daya yang cukup untuk membantu Anda menerapkan TABK. Bagaimanapun, keberhasilan Anda untuk melaksanakan TABK akan sangat bergantung pada dukungan data dan orang-orang terkait. Apabila dukungan ini tidak ada, Anda tentu akan menemui banyak hambatan, yang ujung-ujungnya akan menggagalkan pemenuhan tujuan audit yang Anda tetapkan.

Lalu, dengan melihat berbagai kemungkinan perolehan dan pengujian data, Anda juga harus memutuskan jenis TABK yang sesuai untuk penugasan tersebut. Jika ada beberapa alternatif pilihan TABK, Anda harus membuat perbandingan mengenai kelebihan dan kekurangan masing-masing pendekatan yang didasarkan pada situasi dan kondisi yang Anda hadapi di lingkungan auditee.

Terakhir, seluruh pertimbangan mengenai pemilihan TABK harus didasarkan pada faktor penentu yang berasal dari diri Anda sendiri, yaitu kemampuan personil auditor dan biaya audit yang tersedia. Kemampuan tim audit harus menjadi faktor penentu keberhasilan TABK, terlebih apabila TABK yang Anda pilih ternyata memerlukan tingkatan pengetahuan yang lebih tinggi. Untuk mengatasi hal ini, Anda memang diperkenankan untuk meminta bantuan tenaga ahli di bidang itu. Namun, keputusan untuk melibatkan ahli TABK tentu tidak lepas dari ketersediaan dana untuk penugasan tersebut. Jika dana tidak cukup, Anda tentu perlu berpikir ulang mengenai pilihan TABK untuk penugasan audit yang akan Anda lakukan. Tentu saja, pilihan ini ada konsekuensinya.

2.4 Tahap Pelaksanaan TABK

Setelah melalui berbagai pertimbangan seperti dijelaskan di atas, anggaplah bahwa Anda akhirnya memutuskan untuk memilih satu jenis TABK dalam penugasan audit Anda. Untuk memulai menjalankan TABK, Anda sebaiknya mengikuti empat tahapan atau langkah berikut.

- Memahami lingkungan PDE
- Menentukan dan meminta data yang akan dianalisis
- Memperoleh data
- Menganalisis data

2.4.1 Memahami Lingkungan PDE

Setiap penugasan audit yang dilakukan dengan TABK harus dimulai dengan upaya untuk memahami lingkungan pengolahan data elektronis yang dimiliki entitas. Hal ini dilakukan karena tidak ada satu entitas pun yang memiliki desain dan penerapan sistem berbasis komputer yang sama persis satu dengan lainnya. Setiap entitas memiliki kekhasan dan untuk alasan ini Anda perlu melakukan orientasi terlebih dahulu sebelum melaksanakan TABK.

Dalam pemahaman lingkungan ini, Anda harus fokus pada dua hal, yaitu lingkungan aplikasi *(application environment)* dan lingkungan database *(database environment)*.

Tabel 2.2 Informasi Terkait Lingkungan PDE

Lingkungan Aplikasi	Lingkungan Database
Input data	Format data
Menu aplikasi	Struktur data
Laporan	Periode data

Terkait lingkungan aplikasi, yang perlu Anda pelajari meliputi hal-hal terkait desain dan pengoperasian aplikasi tersebut. Informasi minimal yang perlu Anda peroleh di sini adalah bagaimana aplikasi itu dijalankan. Salah satu cara yang bisa Anda lakukan dalam hal ini adalah menempatkan diri seolah-olah Anda akan menjadi seorang operator yang memiliki tanggung jawab untuk menjalankan aplikasi tersebut. Untuk memperoleh pemahaman sebagai seorang operator, Anda tentu harus banyak berinteraksi dengan operator aplikasi tersebut. Dalam interaksi ini, senjata utama Anda adalah bertanya. Anda harus mengeksplorasi mengenai sumber data yang menjadi bahan masukan aplikasi tersebut dan bagaimana cara menginputnya sampai akhirnya masukan tadi diproses oleh komputer. Anda juga harus menanyakan berbagai laporan yang dihasilkan oleh aplikasi tersebut dan bagaimana cara membuatnya. Untuk itu, Anda harus mempelajari berbagai menu aplikasi dan alur pengolahan data yang dilakukan oleh aplikasi tersebut.

Setelah paham betul mengenai aplikasi, Anda selanjutnya harus mulai mengeksplorasi berbagai informasi seputar database. Biasanya, pengelolaan database dilakukan tersendiri oleh personel lainnya. Jika memang demikian, Anda harus segera menghubungi personel tersebut untuk memperoleh informasi mengenai database. Kepentingan Anda dengan pengelola database adalah untuk mengetahui tiga hal: format data, struktur data, dan periode data.

Aplikasi dibangun dengan berbagai macam *platform* dan sebagai konsekuensinya, format database yang digunakannya juga akan bermacam-macam. Untuk itu, Anda perlu memperoleh informasi mengenai format data yang diolah oleh aplikasi yang tadi telah Anda pelajari. Terkait format database, Anda hanya diminta untuk memastikan bahwa yang digunakan oleh auditee adalah jenis database yang nantinya mampu Anda olah dengan teknologi yang Anda miliki atau kuasai. Apabila kemudian Anda menjumpai bahwa database yang dimiliki auditee tidak sesuai dengan yang Anda butuhkan, Anda harus memastikan bahwa database tersebut dapat diubah ke format yang Anda minta.

Lalu, sedikit lebih rinci, Anda harus memperoleh informasi mengenai struktur database, yang meliputi *record layout,* daftar tabel, dan deskripsinya. Informasi ini akan Anda gunakan untuk memperoleh gambaran mengenai database dan keterkaitannya

satu sama lain. Dengan informasi ini pula, Anda akan dapat menentukan tabel-tabel mana yang relevan dengan tujuan audit Anda. Berikut ini contohnya.

Tabel 2.3 Struktur Database

Nama *Field*	Panjang	Tipe	Format	Keterangan
KODE_BRG	7	Character		Nomor Produk
NAMA_BRG	20	Character		Keterangan Produk
HJUAL_UNIT	6	Numeric	9,999.99	Harga Jual per Unit
HJUAL_TGL	10	Date	YY/MM/DD	Tanggal Harga Jual

Terakhir, Anda harus menanyakan periode yang tercakup dalam database. Informasi mengenai periode ini tentu saja penting karena akan terkait dengan cakupan penugasan audit yang Anda lakukan. Anda tentu tidak akan meminta semua data yang ada, melainkan hanya data yang masuk dalam cakupan penugasan, misalnya satu tahun, dua tahun, atau berapa pun periode yang telah Anda tetapkan.

2.4.2 Menentukan dan Meminta Data yang Akan Dianalisis

Setelah informasi mengenai lingkungan aplikasi dan lingkungan database diperoleh, langkah berikutnya adalah menentukan data mana saja yang Anda perlukan lalu memintanya untuk keperluan analisis lebih lanjut. Sekali lagi, pemilihan data ini tentu mempertimbangkan tujuan audit yang telah ditetapkan.

Permintaan data hendaknya ditujukan kepada personel yang tepat, seperti pengelola database, dengan formulir permintaan yang di dalamnya berisi mengenai berbagai informasi database yang diminta. Penggunaan formulir ini dilakukan untuk menghindari salah pengertian dan dapat digunakan sebagai alat kendali mengenai kebenaran database yang telah Anda terima nantinya. Berikut contoh formulir permintaan data dimaksud.

Tabel 2.4 Contoh Formulir Permintaan Data

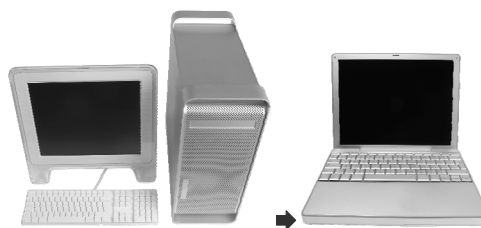
PERMINTAAN DATABASE PERSEDIAAN	
Informasi kontak	
Tanggal permintaan	20 Maret 2011
Kepada	Joni Teknisi, Kantor Pusat PT Autechno
Alamat email	joni_tek@autechno.com
Dari	Anton, Ak
Alamat email	anton@kapmbelgedes.com
Data yang diminta	
Nama file atau data set	persediaan.dbf (transaksi persediaan)
Nama *field* atau tabel	Sesuai saran staf bagian Sistem Informasi
Tanggal data	1 Januari – 31 Desember 2010
Platform dan aplikasi	PC, Oracle Financials
Struktur file	dBase (dbf)
Rincian transfer data	
Data diminta tanggal	23 Maret 2011
Media penyimpanan	CD
Metode transfer	Kurir
Alamat penerima	Kantor Akuntan Publik M. Belge DES, CPA, Jalan Sidobali 100, Jakarta
Informasi tambahan	
Jumlah *record*	Informasikan melalui email
Control total untuk *field numeric*	Informasikan melalui email

2.4.3 Memperoleh Data

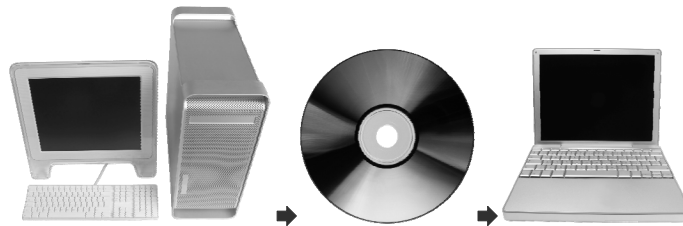
Bersamaan dengan penentuan dan permintaan data, Anda juga harus mempertimbangkan cara perolehan data yang Anda minta, atau dengan kata lain, bagaimana caranya data itu sampai ke tangan Anda. Pilihan cara untuk memperoleh data tidak terlepas dari pertimbangan keamanan dan kemudahan akses data tersebut nantinya. Secara teoritis, proses pemindahan tersebut bisa dilakukan dengan dua cara: langsung dan tidak langsung.

Pemindahan Langsung

Jika memilih metode langsung maka Anda akan mengakses data tersebut langsung dari lokasi di mana data tersebut tersimpan (server). Perlu dicatat bahwa data yang Anda akses dan akan Anda salin secara langsung ini adalah data yang sedang digunakan oleh aplikasi yang diaudit. Metode ini dapat dilakukan melalui jaringan komputer (LAN/WAN) atau langsung di server di mana data tersebut disimpan.

Kelebihan metode ini adalah adanya keleluasan bagi auditor untuk memilih data yang akan digunakan serta akan selalu memperoleh data yang terbaru. Namun perlu diperhatikan bahwa metode ini memiliki pengaruh pada aplikasi yang sedang berjalan. Sebagai contoh, apabila data yang diakses melalui jaringan komputer tersebut memiliki ukuran sangat besar maka proses perolehan data ini akan dapat memengaruhi kecepatan komunikasi komputer-komputer lain yang berada di jaringan yang sama.

Gambar 2.4 Pemindahan Langsung

Pemindahan Tidak Langsung

Sementara itu, dengan metode pemindahan tidak langsung, yang Anda lakukan adalah menyalin data yang akan diaudit dan memindahkannya melalui media sementara ke komputer auditor. Metode ini dapat dilakukan menggunakan media sekunder, seperti *CD-ROM, flash disk, hard disk eksternal, zip drive, tape drive,* atau media lainnnya. Jika dimungkinkan Anda juga dapat menyalin data yang akan diaudit melalui jaringan komputer dan memindahkannya ke komputer auditor. Metode akan menghilangkan risiko munculnya masalah pada jaringan komunikasi ataupun pada aplikasi yang diaudit sendiri jika data diakses secara langsung. Namun, metode ini hanya memberikan auditor data sampai dengan saat data tersebut disalin. Dan jika auditor memerlukan data yang terbaru maka auditor harus melakukan kembali penyalinan data yang terbaru tersebut.

Gambar 2.4 Pemindahan Tidak Langsung

Metode tidak langsung data adalah yang paling banyak dipilih oleh auditor dalam memperoleh akses ke data yang akan dianalisis. Pembahasan pada buku ini juga menggunakan asumsi pemindahan data tidak langsung.

2.4.4 Menganalisis Data

Terakhir, setelah memperoleh data, Anda dapat melakukan serangkaian analisis dan pengujian dalam rangka memenuhi penugasan yang Anda terima. Meskipun berbagai teknik analisis dan pengujian akan dibahas detail dalam buku ini, namun teknik itu tidak dengan sendirinya dapat menyelesaian seluruh tugas Anda. Teknik-teknik analisis itu hanyalah alat bantu yang bisa

bermanfaat apabila digunakan dalam konteks pemenuhan tujuan audit. Sehingga analisis data dengan TABK pasti akan mengikuti tiga tahapan kecil berikut.

- Menetapkan tujuan audit
- Mengakses data
- Menerapkan teknik analisis dan pengujian

Tujuan audit merupakan sebuah alasan mengapa Anda melakukan berbagai prosedur audit. Agar prosedur dapat diterapkan secara efektif, Anda tentu harus memiliki atau merumuskan tujuan audit secara spesifik. Setelah tujuan ditetapkan, Anda lalu dapat memilih data mana yang relevan dengan tujuan tersebut. Dalam hal ini, Anda akan mengakses data yang telah disimpan dalam media penyimpanan tertentu. Lalu, serangkaian teknik analisis dan pengujian data akan Anda eksekusi sebelum akhirnya membuat sebuah kesimpulan.

Sebagai contoh, anggaplah bahwa salah satu tujuan audit yang ingin Anda penuhi adalah menilai kelengkapan transaksi penjualan. Untuk itu, yang Anda butuhkan adalah data penjualan yang terjadi dalam satu tahun fiskal, misalnya dari tanggal 1 Januari s.d. 31 Desember 2010. Setelah memastikan ketersediaan data dimaksud, Anda kemudian mengakses datanya dengan sebuah teknologi yang akan digunakan untuk menganalisis data tersebut. Namun, sebelum Anda mempraktikkan cara analisis data, Anda tentu harus paham dulu mengenai…